ERROR CORRECTION METHOD, ERROR CORRECTION APPARATUS AND ERROR CORRECTION PROGRAM

InsA1>

TECHNICAL FIELD

The present invention generally relates to a method of error correction, an error correction apparatus and a recording medium storing an error correction program, for correcting an error that occurs in digital wireless communication and digital magnetic recording.

BACKGROUND ART

In general, decoding of a BCH code according to the related art comprises the following steps.

I. Calculation of syndrome
II. Estimation of number of bits in error
III. Calculation of error location polynomial
IV. Calculation of root of error location polynomial
V. Correction of error location

Calculation of a root of an error location polynomial is usually performed by executing a method called Chien search for successively substituting an element of a Galois field in the error location polynomial and checking whether that element is the root.

Since the maximum number of steps

required in the Chien search method is the length of a code, an effective, high-speed decoding cannot be performed. In order to solve this problem, a direct solution of an error location polynomial is proposed in the decoding of a triple and quadruple error correcting BCH code.

Direct solutions are disclosed in Hirokazu OKANAO "Decoding of Triple and Quadruple Error Correcting BCH Code by Direct Solving of Error Location Polynomials," Collected Papers, Institute of Electronics and Communication Engineers of Japan (Vol. J64-A, No. 2):137-144 and in Japanese Laid-Open Patent Application No. 59-165153.

The following is a description of a related-art method of error correction using a three-bit correcting BCH code and a four-bit correcting BCH code disclosed in "Decoding of Triple and Quadruple Error Correcting BCH Code by Direct Solving of Error Location Polynomials".

Fig. 1 is a flowchart showing a decoding algorithm of a three-bit correcting BCH code over a Galois field GF(2^N) described in the above-mentioned paper. With reference to the drawing, the operation of the algorithm will be described. In the following description, it is assumed that N is an even number.

First, syndromes S1, S3 and S5 are calculated from a received word (step ST1). If the syndromes S1, S3 and S5 are all 0, it is determined that there is no error and the decoding process is terminated (steps ST2, ST3). When not all of the syndromes S1, S3 and S5 are 0, a calculation $U=S1^3+S3$ is performed. If U=0, an estimation is made that there is a one-bit error (steps ST2, ST4, ST5).

If U is not 0, a calculation

$$V=S1^3+S3+(S1^3\cdot S3+S1\cdot S5)/(S1^3+S3)$$

is performed. If v=0, an estimation is made that there is a two-bit error so that an algorithm for solving a quadratic equation is executed (steps ST6, ST7, ST8). If v is not 0, an estimation is made that there is a three-bit error so that an algorithm for solving a cubic equation is executed (steps ST6, ST9, ST10).

An error location polynomial for one-bit error is given by the following expression (1). The root of expression (1) is S1.

$$x+S1=0 \quad (1)$$

An error location polynomial for a two-bit error is given by the following expression (2).

$$x^2+S1\cdot x+(S1^3+S3)/S1=0 \quad (2)$$

Given below is a description of how to obtain the root of expression (2). For simplicity of the description, a solution of a general quadratic equation such as (3), instead of expression (2), will be described.

$$X^2+\sigma 1\cdot x+\sigma 2=0 \quad (3)$$

The root of expression (3) is $x=\sigma 2^{1/2}$ when $\sigma 1=0$. Expression (3) is transformed into a normalized quadratic equation (4) when $\sigma 1\neq 0$ and defining such that $x=\sigma 1\cdot y$.

$$Y^2+y+\sigma 2/\sigma 1^2=0 \quad (4)$$

One root of expression (4) is determined by storing in the table roots of expression (4) for constant terms ($\sigma 2/\sigma 1^2$) and referring to the table with the constant term. When one root stored in the table is designated as y1, the other root is determined as y1+1 according to the relation between the solution and the coefficient. The two roots $x1=\sigma 1\cdot y1$ and $x2=x1+\sigma 1$ of expression (3) are determined from the two roots y1 and y1+1 of expression (4).

Fig. 2 is a flowchart showing a solution algorithm for solving a quadratic equation $x^2+\sigma 1\cdot x+\sigma 2=0$. Correction becomes impossible when $\sigma 1=0$ since an adequate error location polynomial does not have multiple roots.

The error location polynomial for a three-bit error is given in general by expression (5). The coefficient of expression (5) is calculated by the syndromes.

$$X^3+\sigma 1\cdot x^2+\sigma 2\cdot x+\sigma 3=0 \quad (5)$$

Expression (5) is transformed into a normalized cubic equation (6) when defining such that $x=y+\sigma 1$.

$$y^3+p\cdot y+q=0 \quad (6)$$

$$P=\sigma 1^2+\sigma 2$$

$$q=\sigma 1\cdot \sigma 2+\sigma 3$$

In general, assuming that ω is a cubic root of 1 (the cubic root of 1 exists since we assume that N of the Galois field $GF(2^N)$ to be an even number), the following expression results.

$$\{y+(\beta+\gamma)\}\{y+(\beta\cdot\omega+\gamma\cdot\omega^2)\}$$

$$\{y+(\beta\cdot\omega^2+\gamma\cdot\omega)\}=y^3+\beta\cdot\gamma\cdot y+\beta^3+\gamma^3 \quad (7)$$

Equating the corresponding coefficients of expression (7) and expression (6), the following two relational expressions (expression (8)) are obtained.

$$\beta^3+\gamma^3=q$$

$$\beta\cdot\gamma=p \quad (8)$$

Expression (8) shows that β^3 and γ^3 are two roots of the following quadratic equation (9).

$t^2+q\cdot t+p^3=0$ (9)

Expression (9) is solved by using the method of solving the quadratic equation mentioned above. $t1=\beta^3$ and $t2=\gamma^3$ when the two roots are designated as t1 and t2, and, from this, β and γ are obtained by referring to the cubic root table.

There are three cubic roots but if one of these is stored in the table, the other two roots are obtained by using the cubic root ω of 1. If $t1^{1/3}$ is obtained by referring to the cubic root table with t1, β is either $t1^{1/3}$, $t1^{1/3}\cdot\omega$ or $t1^{1/3}\cdot\omega^2$. If $t2^{1/3}$ is obtained by referring to the cubic root table with t2, γ is either $t2^{1/3}$, $t2^{1/3}\cdot\omega$ or $t2^{1/3}\cdot\omega^2$.

Here, three roots $y1=\beta+\gamma$, $y2=\beta\cdot\omega+\gamma\cdot\omega^2$ and $y3=\beta\omega^2+\gamma\cdot\omega$ of expression (6) are obtained when β and γ are chosen to satisfy the second expression of expression (8). In addition, three roots $x1=y1+\sigma1$, $x2=y2+\sigma1$ and $x3=y3+\sigma1$ of expression (5) are obtained.

Fig. 3 is a flowchart showing a cubic equation solution algorithm. Fig. 4 is a flowchart showing a solution algorithm for solving the normalized cubic equation $y^3+py+q=0$ of Fig. 3. The cubic root table is necessary in this algorithm as shown in Fig. 4.

The root is calculated by the above-described method of directly solving the error location polynomial when it is determined that there is a one-bit to three-bit error. Since the exponent of the calculated root (element of a Galois field) corresponds to an error location, an error location is identified for correction, by preparing a table that stores an exponent for each element of the Galois field and by referring to the table with the element of the Galois field.

When the error location is identified, the error at the error location is corrected (step ST11), so that the decoding result is output.

As described above, the algorithm becomes complex in the decoding of a three-bit correcting BCH code according to the related art since separate processes are required for a two-bit error and a three-bit error. In addition, the root table of the normalized quadratic equation to solve the quadratic equation and the cubic root table to solve the normalized cubic equation are necessary in direct solution of the error location polynomial. Moreover, a table which maps the calculated root to the error location is necessary so that a large storage capacity for storing these tables is necessary.

A description will be given of the error correction method using a four-bit correcting BCH code.

Fig. 5 is a flowchart showing a decoding algorithm described in the above-mentioned paper for decoding a four-bit correcting BCH code over a Galois field GF(2^N). With reference to the drawing, the operation of the algorithm will be described.

In the following description, it is assumed that N is an even number.

First of all, syndrome S1, S3, S5, and S7 are calculated from the received word (step ST21). It is determined that there is no error if all of the syndromes S1, S3, S5, and S7 are 0. The decoding is then terminated (steps ST22 and ST23).

When not all of the syndromes S1, S3, S5, and S7 are 0, $U=S1^3+S3$ and $V=S1(S1^5+S5)+S3(S1^3+S3)$ are calculated. It is determined that an error of two bits or fewer occurred when V=0. It is determined that a one-bit error occurred in case of U=0 (steps ST24, ST25, and ST26), and it is determined that a two-bit error occurred when U≠0 (steps ST24, ST25, and ST27).

It is assumed that a three-bit error or a four-bit error occurred when V≠0.

The error location polynomial for a one-bit error and a two-bit error is the same as the one described in the error correction method using the three-bit correcting BCH code mentioned above (steps ST26, ST27, and ST28). When a three-bit error or a four-bit error occurs, the coefficient of the quartic error location polynomial (10) shown below is calculated (step ST29).

$$X^4+\sigma 1\cdot x^3+\sigma 2\cdot x^2+\sigma 3\cdot x+\sigma 4=0 \quad (10)$$

It is assumed here that there is a three-bit error if the constant term σ4 is 0. In this case, the cubic equation solution algorithm is executed (steps ST30 and ST31). When the constant term σ4 is not 0, an assumption is made that there is a four-bit error. In this case, the quartic equation solution algorithm is executed (steps ST30 and ST32).

For a three-bit error, the error location polynomial is a cubic equation (11) shown below. It is possible to solve this equation by using the cubic equation solution algorithm given in the description of the decoding of the three-bit correcting BCH code (step ST31).

$$X^3+\sigma 1\cdot x^2+\sigma 2\cdot x+\sigma 3=0 \quad (11)$$

Next, a description will be given of the solution of a quartic equation for a four-bit error.

When factorized into a product of quadratic polynomials, expression (10) is represented as expansion (12) below.

$$(x^2+p1\cdot x+q1)(x^2+p2\cdot x+q2)$$
$$=x^4+(p1+p2)x^3+(q1+q2+p1\cdot p2)x^2$$
$$+(p1\cdot q2+p2\cdot q1)x+q1\cdot q2 \quad (12)$$

Equating the corresponding coefficients of expressions (10) and (12), we obtain the following four relations (expression (13)).

$$P1+p2=\sigma 1$$

$q1+q2+p1\cdot p2=\sigma2$

$P1\cdot q2+p2\cdot q1=\sigma3$

$q1\cdot q2=\sigma4$ (13)

A normalized cubic equation (14) is derived from expression (13) by defining such that $p1\cdot p2=\lambda$.

$$\lambda^3+(\sigma2^2+\sigma1\cdot\sigma3)\lambda+\sigma1\cdot\sigma2\cdot\sigma3+\sigma3^2+\sigma1^2\cdot\sigma4=0 \quad (14)$$

By solving expression (14) by the above-mentioned solution algorithm for solving a normalized cubic equation, and given that one root is λ1, p1 and p2 are obtained as two roots of the quadratic equation (15), and q1 and q2 are obtained as two roots of the quadratic equation (16).

$$X^2+\sigma1\cdot x+\lambda1=0 \quad (15)$$

$$X^2+(\sigma2+\lambda1)x+\sigma4=0 \quad (16)$$

(p1,q1) and (p2,q2) are chosen to satisfy the third expression of expression (13). Solving the quadratic equation (17) with p1 and q1 given, the two roots x1 and x2 of expression (10) are obtained. Solving the quadratic equation (18) with p2 and q2 given, the remaining two roots x3 and x4 of expression (10) are obtained.

$$x^2+p1\cdot x+q1=0 \quad (17)$$

$$x^2+p2\cdot x+q2=0 \quad (18)$$

Fig. 6 is a flowchart showing the solution algorithm for solving a quartic error location polynomial. As shown in the figure, it is necessary to solve the quadratic equation four times in this algorithm. Moreover, the process is complex and is of a large-scale in that a suitable combination must be chosen from p1, p2, q1, and q2.

The roots are calculated by the above-mentioned direct solutions of the error location polynomial when it is determined that there is an error of one to four bits. Since the exponent of the calculated root (element of the Galois field) shows the error location, the error location is identified by referring to the table.

As described above, the algorithm for decoding the four-bit correcting BCH code is complex since separate processes are required for a three-bit error and a four-bit error. The algorithm for direct solution of the quartic error location polynomial is complex and a large number of processing steps are required, since it is necessary to solve a quadratic equation four times. Moreover, the root table of normalized quadratic equation to solve a quadratic equation and the cubic root table to solve a normalized cubic equation are necessary, as in the case of the three-bit correcting BCH code. Moreover, a table which maps the calculated root of the error location polynomial to the error location is necessary so that a large storage capacity to store these tables is required. With this construction, the related-art error correction method is complex and requires a large volume of processes in that it requires frequent branching of processes for the decoding algorithm. Another problem is that a large storage capacity is necessary to prepare a lot of tables.

The above-mentioned tables are contained in ROMs when the related-art algorithm is implemented by a circuit. However, there is a problem in that abundant use of ROMs invites a large circuit scale.

Accordingly, an object is to provide an error correction method, an error correction apparatus and an error correction computer program capable of simplifying a decoding algorithm so that the volume of processes is reduced.

Another object is to provide an error correction method, an error correction apparatus and an error correction computer program in which the table size is reduced so that the storage capacity is reduced.

DISCLOSURE OF THE INVENTION

The aforementioned objects are achieved by an error correction method comprising: a first step of calculating syndromes from a received word and estimating the number of bits in error from the syndromes; a second step of generating a cubic error location polynomial from the syndromes, when it is determined that there is a two-bit error or a three-bit error; a third step of determining a normalized cubic equation from the cubic error location polynomial, calculating roots of the normalized cubic equation, and calculating roots of the cubic error location polynomial from the roots of the normalized cubic equation; and a fourth step of identifying an error location from the roots of the cubic error location polynomial and correcting a value of information bit of the error location.

With this, the decoding algorithm is simplified and the volume of processes is reduced.

The third step may further comprise the steps of: translating the error location polynomial over a Galois field into a polynomial over a subfield, calculating a cubic root in the subfield, and calculating a cubic root in the Galois field from the cubic root in the subfield, so as to calculate the roots of the normalized cubic equation.

With this, the table size is reduced and the storage capacity is reduced.

The fourth step may comprise the steps of substituting a root of the error location polynomial for a Galois field element and determining the Galois field element corresponding to the error location by cyclic steps of comparison as the Galois field element is multiplied by a predetermined coefficient at each step.

With this, the Galois field table for

calculating the error location from the roots of the error location polynomial becomes unnecessary and the required table size is reduced.

The aforementioned objects are also achieved by an error correction method comprising: a first step of calculating syndromes from a received word and estimating the number of bits in error from the syndromes; a second step of generating a quadratic error location polynomial or a quartic error location polynomial depending on the number of bits in error estimated by the first step; a third step of calculating roots of the quadratic error location polynomial generated in the second step; a fourth step of calculating roots of the quartic error location polynomial generated in the second step; and a fifth step of identifying an error location, based on the roots of the quadratic error location polynomial calculated in the third step or the roots of the quartic error location polynomial calculated in the fourth step, and correcting a value at the error location.

With this, the decoding algorithm is simplified and the volume of processes is reduced.

The fourth step may comprise: a sixth step of generating a normalized cubic equation from the quartic error location polynomial generated in the second step, and calculating roots of the normalized cubic equation; a seventh step of generating a quadratic equation from the normalized cubic equation calculated from the roots of the normalized cubic equation calculated in the sixth step, and calculating roots of the quadratic equation; an eighth step of generating a pair of two quadratic equations from the roots of quadratic equation calculated in the seventh step, and calculating four roots of the pair of qudratic equations; and a ninth step of identifying the roots of the quartic error location polynomial from the four roots of the pair of quadratic equations calculated in the eighth step.

With this, the table size is reduced and the storage capacity is reduced.

The sixth step may calculate the roots of the normalized cubic equation, by translating a polynomial of the normalized cubic equation polynomial over a Galois field into a polynomial over a subfield, and calculating a cubic root in the Galois field from the cubic root in the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The aforementioned objects are also achieved an error correction method comprising the steps of: a first step of performing arithmetic operations in a subfield of a Galois field so as to calculate syndromes from a received word, and estimating the number of bits in error from the syndromes; a second step of generating an error location polynomial in accordance with the number of bits in error estimated by the first step; a third step of calculating roots of the error location polynomial generated by the second step: and a fourth step of identifying an error location from the roots of the error location polynomial calculated in the third step, and correcting a value of information bit at the error location.

With this, the error location is identified and the value at the error location is corrected efficiently. In addition, the table size is reduced and the storage capacity is reduced.

The first step may use an exponential representation to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The first step may use a vector representation to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The first step may use a normalized basis to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The first step may use a dual basis to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The aforementioned objects are also achieved by an error correction apparatus comprising: error bit count estimating means for calculating syndromes from a received word and estimating the number of bits in error from the syndromes; polynomial generating means for generating a quadratic error location polynomial or a quartic error location polynomial depending on the number of bits in error estimated by the error bit count estimating means; polynomial solution means for determining a normalized cubic equation from the cubic error location polynomial, calculating roots of the normalized cubic equation, and calculating roots of the cubic error location polynomial from the roots of the normalized cubic equation; and correcting means for identifying an error location from the roots of the cubic error location polynomial and correcting a value of information bit of the error location.

With this, the decoding algorithm is simplified and the volume of processes is reduced.

The polynomial solution means may calculate the roots of the normalized cubic equation, by translating the error location polynomial over a Galois field into a polynomial over a subfield, calculating a cubic root in the subfield, and calculating a cubic root in the Galois field from the cubic root in the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The correcting means may identify the error location, by substituting a root of the error location polynomial for a Galois field element and determining the Galois field element corresponding to the error location by cyclic steps of comparison as the Galois field element is multiplied by a predetermined coefficient at each step.

With this, the Galois field table for calculating the error location from the roots of the error location polynomial becomes unnecessary and, as a result, the required table size is reduced and the circuit scale is reduced.

There may be provided a plurality of error correcting means for identifying the error location, by substituting a root of the error location polynomial for a Galois field element and determining the Galois field element corresponding to the error location by cyclic steps of comparison as the Galois field element is multiplied by a predetermined coefficient at each step.

With this, the speed of the process is improved.

The aforementioned objects are also achieved an error correction apparatus comprising: error bit estimating means for calculating syndromes from a received word and estimating the number of bits in error from the syndromes; polynomial generating means for generating a quadratic error location polynomial or a quartic error location polynomial depending on the number of bits in error estimated by the error bit estimating means; qudratic equation solution means for calculating roots of the quadratic error location polynomial generated by the polynomial generating means; quartic equation solution means for calculating roots of the quartic error location polynomial generated by the polynomial generating means; and error correcting means for identifying an error location, based on the roots of the quadratic error location polynomial calculated by the quadratic equation solution means or the roots of the quartic error location polynomial calculated by the quartic equation solution means, and correcting a value at the error location.

With this, the decoding algorithm is simplified and the volume of processes is reduced.

The quartic equation solution means may comprise: cubic equation solution means for generating a normalized cubic equation from the quartic error location polynomial generated by the polynomial generating means, and calculating roots of the normalized cubic equation; first quadratic equation solution means for generating a quadratic equation from the normalized cubic equation calculated from the roots of the normalized cubic equation calculated by the cubic equation solution means, and calculating roots of the quadratic equation; second quadratic equation solution means for generating a pair of two quadratic equations from the roots of quadratic equation calculated by the first quadratic equation solution means, and calculating four roots of the pair of qudratic equations; and root identifying means for identifying the roots of the quartic error location polynomial from the four roots of the pair of quadratic equations calculated by the second quadratic equation solution means.

With this, the table size is reduced and the storage capacity is reduced.

The cubic equation solution means may calculate the roots of the normalized cubic equation, by translating a polynomial of the normalized cubic equation polynomial over a Galois field into a polynomial over a subfield, and calculating a cubic root in the Galois field from the cubic root in the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The aforementioned objects are also achieved by an error correction apparatus comprising: error bit count estimating means for performing arithmetic operations in a subfield of a Galois field so as to calculate syndromes from a received word, and estimating the number of bits in error from the syndromes.

With this, the error location is identified and the value at the error location is corrected efficiently. In addition, the table size is reduced and the storage capacity is reduced.

The error bit count estimating means may use an exponential representation to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The error bit estimating means uses a vector representation to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The error bit count estimating means may use a normalized basis to represent the subfield.

With this, the table size is reduced and the storage capacity is reduced.

The aforementioned objects are also achieved by a recording medium storing an error correction computer program product including steps for: calculating syndromes from a received word and estimating the number of bits in error from the syndromes; determining a normalized cubic equation from the cubic error location polynomial, calculating roots of the normalized cubic equation, and calculating roots of the cubic error location polynomial from the roots of the normalized cubic equation; and determining a normalized cubic equation from the cubic error location polynomial, calculating roots of the normalized cubic equation, and calculating roots of the cubic error location polynomial from the roots of the normalized cubic equation; and identifying an error location from the roots of the cubic error location polynomial and correcting a value of information bit of the error location.

With this, the decoding algorithm is simplified and the volume of processes is reduced.

The aforementioned objects are also

achieved by a recording medium storing an error correction program product including steps for: calculating syndromes from a received word and estimating the number of bits in error from the syndromes; generating a quadratic error location polynomial or a quartic error location polynomial depending on the number of bits in error; calculating roots of the quadratic error location polynomial; calculating roots of the quartic error location polynomial; and identifying an error location, based on the roots of the quadratic error location polynomial calculated in the third step or the roots of the quartic error location polynomial, and correcting a value at the error location.

With this, the decoding algorithm is simplified and the volume of processes is reduced.

The aforementioned objects are also achieved by a recording medium storing an error correction program product including steps for: performing arithmetic operations in a subfield of a Galois field so as to calculate syndromes from a received word, and estimating the number of bits in error from the syndromes.

With this, the error location is identified and the value at the error location is corrected efficiently. In addition, the table size is reduced and the storage capacity is reduced.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a flowchart showing an algorithm of decoding a three-bit correcting BCH code over a Galois field GF(2^N);

Fig. 2 is a flowchart showing a solution algorithm for solving a quadratic equation $x^2+\sigma1\cdot x+\sigma2=0$;

Fig. 3 is a flowchart showing a cubic equation solution algorithm;

Fig. 4 is a flowchart showing a solution algorithm for solving a normalized cubic equation $y^3+py+q=0$;

Fig. 5 is a flowchart showing a decoding algorithm of a four-bit correcting BCH code over a Galois field GF(2^N);

Fig. 6 is a flowchart showing a solution algorithm for solving a quartic error location polynomial;

Fig. 7 is a flowchart showing an error correction method using a three-bit correcting BCH code according to a first embodiment the present invention;

Fig. 8 is a flowchart showing a cubic root calculation algorithm;

Fig. 9 is a flowchart showing an error correction method using a four-bit correcting BCH code according to a second embodiment the present invention;

Fig. 10 is a flowchart showing a quartic equation solution algorithm according to the second embodiment;

Fig. 11 is a flowchart showing an error location calculation algorithm according to a third embodiment of the present invention;

Fig. 12 is a block diagram showing an error correction apparatus using a three-bit correcting BCH code according to a fourth embodiment the present invention;

Fig. 13 is a block diagram showing a quadratic equation solution circuit;

Fig. 14 is a block diagram showing a cubic root calculation circuit;

Fig. 15 is a block diagram showing a translating circuit;

Fig. 16 is a block diagram showing a normalized cubic equation solution circuit;

Fig. 17 is a block diagram showing a translating circuit;

Fig. 18 is a block diagram showing an error location polynomial solution circuit;

Fig. 19 is a block diagram showing an error location polynomial solution circuit;

Fig. 20 is a block diagram showing a quartic equation solution circuit;

Fig. 21 is a block diagram showing an error correction apparatus according to a sixth embodiment of the present invention;

Fig. 22 is a block diagram showing an error location detecting circuit;

Fig. 23 is a flowchart of an error correction method using a three-bit correcting BCH code;

Fig. 24 is a flowchart showing a

computing method of a syndrome S1;

Fig. 25 is a block diagram showing an error correction apparatus using a one-bit correcting BCH code;

Fig. 26 is a block diagram showing an error correction apparatus according to a tenth embodiment of the present invention;

Fig. 27 is a flowchart for calculating a product X·Y of elements X=(x1,x0) and Y=(y1,y0) of a Galois field K;

Fig. 28 is a flowchart for calculating an inverse element X^{-1} of an element X=(x1,x0) of a Galois field K;

Fig. 29 is a block diagram of a Galois field operational processor;

Fig. 30 is a block diagram showing an error correction apparatus according to an eleventh embodiment of the present invention; and

Fig. 31 is a block diagram of a syndrome generating circuit.

BEST MODE OF CARRYING OUT THE INVENTION

Hereinafter, the best mode of carrying out the invention will be described with reference to the attached drawings.

First embodiment

Fig. 7 is a flowchart showing an error correction method using a three-bit correcting BCH code according to a first embodiment the present invention. Referring to Fig. 7, ST51 indicates a step for calculating syndromes S1, S2 and S3. ST52 indicates a step for determining whether there is an error. ST53 indicates a step for terminating the process upon determination that there is no error. ST54 indicates a step for determining whether there is a one-bit error. Steps ST51, ST52 and ST54 constitute an error bit estimation step.

ST55 indicates a step for correcting a one-bit error. ST56 indicates a step for generating a cubic error location polynomial from the syndromes. ST57 indicates a step for determining a normalized cubit equation from the cubic error location polynomial, calculating the root of the normalized cubic equation, and calculating the root of the cubic error location polynomial with the root of the normalized cubic equation being given. ST58 indicates a step for identifying the error location with the root of the cubic error location polynomial being given and correcting the bit value at the error location.

A description will now be given of the operation according to the first embodiment.

A detailed description will be given using a (n, k) three-bit correcting BCH code over a Galois field GF(2^8) having a code length of n for k symbols.

First of all, syndromes S1, S3, and S5 are calculated from the received word in step ST51. When the received bits (r_{n-1}, r_{n-2}, ...,r_1, r_0) are represented by a polynomial (21), the syndromes are calculated such that $S1=R(\alpha)$, $S3=R(\alpha^3)$ and $S5=R(\alpha^5)$. α is a primitive element of the Galois field and is the root of a primitive polynomial $x^8+x^4+x^3+x^2+1$.

$$R(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+ \ldots +r_2x^2+r_1x+r_0 \quad (21)$$

Next, it is determined in step ST52 that there is no error if all of the syndromes S1, S3, and S5 are 0. The decoding process is then terminated (step ST53). When not all of the syndromes S1, S3 and S5 are 0, a calculation $T=S1^3+S3$ is performed in step ST54. If T=0, an estimation is made that there is a one-bit error (step ST55).

In this case, the element of the Galois field corresponding to the error location is S1. Since the exponent of an element of a Galois field corresponds to an error location, an error location is identified for correction, by preparing a table that stores an exponent for each element of the Galois field and by referring to the table with the element of the Galois field.

When T is not 0, the cubic error location polynomial (22) is generated in step ST56. The coefficient of the cubic error location polynomial is calculated from the syndrome.

$$X^3+\sigma1\cdot x^2+\sigma2\cdot x+\sigma3=0 \quad (22)$$

$$\sigma1=S1$$

$$\sigma2=S1^2+(S1^5+S5)/(S1^3+S3)$$

$$\sigma3=S3+S1(S1^5+S5)/(S1^3+S3)$$

In step ST57, the root of the third error location polynomial generated in step ST56 is calculated.

For solution of the cubic equation, the cubic equation solution algorithm and the solution algorithm for solving a normalized cubic equation described as a related-art technology may be employed. In the first embodiment, the quadratic equation solution algorithm and the cubic root calculation algorithm are implemented as described below instead of a lookup table.

First of all, a description will be given of the quadratic equation solution algorithm.

A general quadratic equation is given by expression (23). Expression (23) is transformed into a normalized expression (24) by defining such that $x=p \cdot y$.

$x^2+p \cdot x+q=0$ (23)

$y^2+y+c=0,\ c=q/p^2$ (24)

When polynomial bases $\{\alpha^7, \alpha^6, \alpha^5, \alpha^4, \alpha^3, \alpha^2, \alpha, 1\}$ are given as bases of a Galois field $GF(2^8)$, and c is subject to base expansion as shown in expression (25), one root y of expression (24) is given by expression (26). The other root of expression (24) is determined as y+1, based on a relation between the solution and the coefficient. Thus, two roots $x1=p \cdot y$ and $x2=x1+p$ of expression (23) are obtained.

$$C=c7\cdot\alpha^{7}+c6\cdot\alpha^{6}+c5\cdot\alpha^{5}+c4\cdot\alpha^{4}+c3\cdot\alpha^{3}+c2\cdot\alpha^{2}+c1\cdot\alpha+c0 \quad (25)$$

$$Y=y7\cdot\alpha^{7}+y6\cdot\alpha^{6}+y5\cdot\alpha^{5}+y4\cdot\alpha^{4}+y3\cdot\alpha^{3}+y2\cdot\alpha^{2}+y1\cdot\alpha+y0 \quad (26)$$

$$y7=c0+c1+c2+c4$$

$$y6=c0+c1+c2+c4+c7$$

$$y5=c1+c2+c3+c4+c6$$

$$y4=c0+c7$$

$$y3=c1+c2+c3+c4$$

$$y2=c0+c3+c4+c6$$

$$y1=c0+c2+c4$$

$$y0=0$$

Next, a description will be given of the cubic root calculation algorithm.

In this algorithm, the cubic root of an element of the Galois field $GF(2^{8})$ (hereafter, referred to as K) is calculated by an operation over a subfield $GF(2^{4})$ (hereafter, referred to as L) and using a table.

The base over the subfield L of the Galois field K is given as $\{1, \beta\}$. β is an element of the Galois field K which does not belong to the subfield L, and satisfies $\beta^{2}+p\cdot\beta+q=0$ (p and q are elements of L). In general, the element B of the Galois field K is given by $B=b1\cdot\beta+b0$, using elements b0, b1 of the subfield L.

In the following, it is assumed that b1 is not 0, and a description will be given of the method of calculating the cubic root A of B satisfying $A^{3}=B$.

The above-mentioned equation is transformed into the equation over the subfield, that is, expression (27) and expression (28) by defining such that the cubic root $A=a1\cdot\beta+a0$.

$$(p^2+q)a1^3+p\cdot a0\cdot a1^2+a0^2\cdot a1=b1 \quad (27)$$

$$a0^3+p\cdot q\cdot a1^3+q\cdot a0\cdot a1^2=b0 \quad (28)$$

Expression (29) below is obtained as expression (27) × b0 + expression (28) × b1.

$$\{(p^2+q)b0+p\cdot q\cdot b1\}a1^3$$
$$+(p\cdot b0+q\cdot b1)a0\cdot a1^2$$
$$+b0\cdot a0^2\cdot a1+b1\cdot a0^3$$
$$=0 \quad (29)$$

The cubic equation over the subfield (30) is obtained by dividing both sides of expression (28) by $a1^3$ and defining such that $x=a0/a1$.

$$b1\cdot x^3+b0\cdot x^2+(p\cdot b0+q\cdot b1)x$$
$$+p\cdot q\cdot b1+(p^2+q)b0=0 \quad (30)$$

In addition, the normalized cubic equation over the subfield (31) is obtained by substituting $x=y+b0/b1$ in expression (30).

$$y^3+b2\cdot y+p\cdot b2=0 \quad (31)$$
$$b2=(b0/b1)^2+p(b0/b1)+q$$

Expression (31) is an equation determined by the subfield element b2. By storing one root of the equation (31) in the table for b2, one root of expression (31) is obtained by referring to the table with b2. Therefore, one root x of expression (30) is obtained. $a1^3$ is calculated as shown in expression (32), by using expression (27) so that a1 is obtained by referring to the cubic root table of the subfield. Moreover, a0 is also obtained since a0=x·a1.

$$a1^3=b1/(x^2+p\cdot x+p^2+q) \quad (32)$$

Described above is a method of calculating the cubic root when b1 is not 0. When b1 is 0, B is an element of the subfield so that the cubic root is calculated by referring to the cubic root table of the subfield.

Fig. 8 is a flowchart showing a cubic root calculation algorithm.

The table A referred to in step ST64 is the root table of expression (31). The table B referred to in steps ST67 and ST70 is a cubic root table of the subfield. A description will now be given of the operation with reference to the drawings.

An initial value B is split to fit the subfield in step ST61. Step ST62 examines whether b1 is 0. The algorithm proceeds to step ST70 when b1 is 0, and proceeds to step ST63 when b1 is not 0.

In step ST70, the cubic root q of b0 is output by referring to the table B. When the cubic root q is not found, the algorithm is terminated. When the cubic root q is found, q is substituted for the cubic root A and process is terminated (steps ST71 and ST72).

When it is determined that b1 is not 0 in step ST62, b2 of expression (31) is calculated in step ST63. In step ST64, the root y of expression (31) is output by referring to the table A.

The algorithm proceeds to step ST66 when the root y is found, and when the root y is not found, the process is terminated (step ST65).

In step ST66, the root x of expression (30) is calculated, and, using this, b3, which corresponds to $a1^3$, is calculated. In step ST67, the cubic root q of b3 is calculated by referring to the table B. If the cubic root q is found, the algorithm proceeds to step ST69, and if the cubic root q is not found, the process is terminated (step ST68). In step ST69, a0 is calculated and the cubic root of B is split to fit the subfield before being output.

A concrete example of the necessary size of the tables A and B will be explained. By defining such that $\gamma=\alpha^{17}$, the subfield $GF(2^4)$ is generated. That is, a set $\{0, 1, \gamma, \gamma^2, \ldots, \gamma^{13}, \gamma^{14}\}$ in the Galois field is closed under addition, subtraction, multiplication and division. Defining such that $\beta=\alpha^{123}$, β does not belong to this subfield, and satisfies $\beta^2+p\cdot\beta+q=0\,(p=1, q=\gamma^3)$. The root table of expression (31) (table A) is as large as 8 words by 4 bits, and the cubic root table of the subfield (table B) is as large as 5 words by 4 bits.

The normalized cubic equation is solved by using the quadratic equation solution algorithm and the cubic root calculation algorithm explained above.

Hereafter, a description will be given of the solution of the cubic error location polynomial (22). By defining such that $x=y+\sigma 1$ in expression (22), expression (22) is translated to a normalized cubic equation derived from expression (6) described in the related art. In addition, the quadratic equation (9) is determined by using coefficients p and q of the normalized cubic equation. The quadratic equation is solved by calculating one root t using the quadratic equation solution algorithm described above.

Next, the cubic root β of t is calculated by using the cubic root calculation algorithm mentioned above. γ is given as p/β from expression (8), and three roots of the normalized cubic equation (6) are calculated as shown in expression (33), where ω indicates a cubic root of one. Three roots $x1=y1+\sigma 1$ and $x2=y2+\sigma 1$ and $x3=y3+\sigma 1$ of the cubic error location polynomial (22) are calculated from three the roots of expression (33).

$Y1=\beta+p/\beta$

$Y2=\beta\cdot\omega+p/(\beta\cdot\omega)$

$Y3=\beta\cdot\omega^2+p/(\beta\cdot\omega^2)$ (33)

In step ST58, the error location is identified by referring to the error location table of the Galois field with three roots calculated in step ST57. The error thus identified is then corrected. Expression (22) has a trivial root 0 for a two-bit error. No error correction process is not performed for this root.

According to the first embodiment, the root table of a normalized quadratic equation and the cubic root table to calculate the root of a cubic error location polynomial are unnecessary as described above. By using the table over the subfield, the table size is reduced as compared to the related art. It is also to be noted that, in the related error correction method, a distinction is made between two-bit error correction and three-bit error correction. An advantage provided by the first embodiment is that, by processing two-bit error correction and three-bit error correction using the same sequence, the algorithm is simplified.

In the related-art error correction method, the root table of a normalized quadratic equation is as large as 256 words by 8 bits, the cubic root table 85 words by 8 bits, and the error location table 256 words by 8 bits. In contrast, the table A according to the first embodiment is 8 words by 4 bits, the table B is 5 words by 4 bits and the error location table is 256 words by 8 bits. Excluding the error location table, the table size is approximately 2% that of the related-art method.

Second embodiment

Fig. 9 is a flowchart showing an error correction method using a four-bit correcting BCH code according to a second embodiment the present invention. Referring to Fig. 9, ST81 indicates a step for calculating syndromes S1, S3, S5 and S7. ST82 indicates a step for determining whether there is an error. ST83 indicates a step for terminating the process upon determination that there is no error. ST84 indicates a step for determining whether an error of two bits or fewer occurs. Steps ST81, ST82 and ST84 constitute an error bit estimation step.

ST85 indicates a step for generating a quadratic error location polynomial when an error of two bits or fewer occurs (polynomial generation step), ST86 indicates a step for executing an algorithm for solving the quadratic equation so as to calculate the root of the quadratic error location polynomial. ST87 indicates a step for generating a quartic equation when an error of three bits or more occurs. ST88 indicates a step for executing an algorithm for solving the quartic equation so as to calculate the root of the quartic error location polynomial. ST89 indicates a step for identifying an error location based on the root of the quadratic error location polynomial and the quartic error location polynomial, and correcting the bit value at the error location.

A description will now be given of the operation according to the second embodiment.

A detailed description will be given using a (n, k) four-bit correcting BCH code over a Galois field GF(2^8) having a code length of n for k symbols.

First of all, syndrome S1, S3, S5 and S7 are calculated from the received word in step ST81. When the received bits (r_{n-1}, r_{n-2}, ...,r_1, r_0) are represented by a polynomial (41), the syndromes are calculated such that S1=R(α), S3=R(α^3), S5=R(α^5) and S7=R(α^7). α is a primitive element of the Galois field and is the root of a primitive polynomial $x^8+x^4+x^3+x^2+1$.

$$R(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+\ldots+r_2x^2+r_1x+r_0 \quad (41)$$

Next, it is determined in step ST82 that there is no error if all of the syndromes S1, S3, S5 and S7 are 0. The decoding process is then terminated (step ST83). When not all of the syndromes S1, S3, S5 and S7 are 0, a calculation $V=S1(S1^5+S5)+S3(S1^3+S3)$ is performed in step ST54. If V=0, an estimation is made that there is an error of two bits or fewer, and the algorithm proceeds to step ST85. If V is not 0, it is determined that there is an error of three bits or more, and the algorithm proceeds to step ST87.

In step ST85, the coefficient of the error location polynomial (42) for an error of two bits or fewer is calculated. For a one-bit error, the constant term σ2 is 0 so that expression (42) has a trivial root 0.

$$X^2+\sigma 1 x+\sigma 2=0 \quad (42)$$

$$\sigma 1=S1$$

$$\sigma 2=(S1^3+S3)/S1$$

In step ST86, by executing the quadratic equation solution algorithm described in the first embodiment, the quadratic equation (42) is solved by calculating two roots thereof.

When it is determined in step ST84 that V is not 0, the coefficient of the quartic error location polynomial (43) is calculated in step ST87. In the case of a three-bit error, the constant term σ4 is 0 and expression (43) has a trivial root 0.

$$X^4+\sigma 1\cdot x^3+\sigma 2\cdot x^2+\sigma 3\cdot x+\sigma 4=0 \quad (43)$$

$$\sigma 1=S1$$

$$\sigma 2=\{S1(S1^7+S7)+S3(S1^5+S5)\}$$

$$/\{S3(S1^3+S3)+S1(S1^5+S5)\}$$

$$\sigma 3=\{S1(S1^3\cdot S5+S1\cdot S7)+S3(S1^6+S3^2)\}$$

$$/\{S3(S1^3+S3)+S1(S1^5+S5)\}$$

$$\sigma 4=\{S1^3(S1^7+S7)$$

$$+S3(S1^7+S1\cdot S3^2+S7)$$
$$+S5(S1^5+S1^2\cdot S3+S5)\}$$
$$/\{S3(S1^3+S3)+S1(S1^5+S5)\}$$

Expression (43) is transformed into expression (44) when it is defined such that $x=y+c$ and $c=(\sigma3/\sigma1)^{1/2}$.

$$y^4+p\cdot y^3+q\cdot y^2+r=0 \quad (44)$$
$$P=\sigma1$$
$$Q=\sigma1c+\sigma2$$
$$R=c^4+\sigma1\cdot c^3+\sigma2\cdot c^2+\sigma3\cdot c+\sigma4$$

When factorized into a product of quadratic polynomials, expression (44) is represented as an expansion (45).

$$(y^2+p1y+q1)(y^2+p2y+q2)$$
$$=y^4+(p1+p2)y^3+(q1+q2+p1\cdot p2)y^2$$
$$+(p1\cdot q2+p2\cdot q1)y+q1\cdot q2 \quad (45)$$

Equating the corresponding coefficients of expressions (44) and (45), we obtain the following four relations (expression (46)).

$$p1+p2=p$$
$$q1+q2+p1\cdot p2=q$$
$$p1\cdot q2+p2\cdot q1=0$$
$$q1\cdot q2=r \quad (46)$$

Substituting p1/q1=p2/q2=z in the third expression of expression (46), the first expression is transformed into expression (47) and the second expression is transformed into expression (48).

$z(q1+q2)=p$ (47)

$q1+q2+z^2 \cdot q1 \cdot q2=q$ (48)

A normalized quadratic equation (49) below is obtained by using expressions (47) and (48) and the fourth expression of expression (46).

More specifically, the quadratic equation solution algorithm and the cubic root calculation algorithm described in the first embodiment are used to execute the solution algorithm for solving a normalized cubic equation, so as to calculate one root λ of expression (49). The fourth expression of expression (46) and expression (47) show that q1 and q2 are obtained as two roots of expression (50) also shown below. From this, p1 and p2 are also determined by defining such that $P1=\lambda \cdot q1$ and $p2=\lambda \cdot q2$.

$z^3+(q/r)z+(p/r)=0$ (49)

$t^2+(p/\lambda)t+r=0$ (50)

When p1, p2, q1 and q2 are determined by calculation, the two roots y1, y2 of expression (44) are determined by using expression (51) given below. From expression (52) also given below, the remaining two roots y3, y4 are determined, so that four roots x1=y1+c, x2=y2+c, x3=y3+c, x4=y4+c of expression (43) are determined.

$$y^2+p1\cdot y+q1=0 \quad (51)$$

$$y^2+p2\cdot y+q2=0 \quad (52)$$

Fig. 10 is a flowchart showing a quartic equation solution algorithm according to the second embodiment. ST91 indicates a step for setting a quartic equation, ST92 indicates a step for translating coefficients, ST93 indicates a step for solving a normalized cubic equation, ST94-ST96 indicate steps for solving a quadratic equation and ST97 indicates a step for translating solutions.

A solution algorithm for solving a quartic equation will be described.

In step ST92, p, q, r of expression (44) are calculated from the coefficients of the quartic equation set in step ST91.

In step ST93, one root λ of the normalized cubic equation (49) obtained with p, q, r being given in step ST92.

In step ST94, the two roots q1, q2 of the quadratic equation (50), determined with the root λ and p, r being given as a result of calculation in step ST93, are calculated.

In steps ST95 and ST96, with λ being given as a result of calculation in step ST93 and q1, q2 being given as a result of calculation in step ST94, the two roots of the quadratic equation (51) and the two roots of the quadratic equation (52) are calculated.

In step ST97, the four roots of the quartic equation set in step ST91 are calculated by translating the four roots calculated in steps ST95 and ST96.

In step ST89, the error location is identified based on the root of the error location polynomial calculated in step ST86 or step ST88, so that the error is corrected. Since the exponent of an element of a Galois field corresponds to an error location, an error location is identified for correction, by preparing a table that stores an exponent for each element of the Galois field and by referring to the table with the element of the Galois field. The error at the identified location is corrected so that the decoded result is output. In step ST86, two elements of the Galois field are calculated. In the case of a one-bit error, the zero root not related to the error location is included. In step ST88, four elements of the Galois field are calculated. In the case of three-bit error, zero roots not related to the error location are included. No error correction process is performed for the zero root.

As described, the second embodiment also uses the solution algorithm for solving a normalized cubic equation described in the first embodiment. With this configuration, the root table of the normalized quadratic equation and the cubic root table are unnecessary. By using the table over the subfield, the table size is reduced as compared to the related art. Another advantage is that the error correction method according to the second embodiment only allows branching into an error of two bits or fewer and an error of three bits or greater, in contrast to the related-art error correction method using a four-bit correcting BCH code that allows branching into as many number of processes as the number of bits in error. Accordingly, the algorithm is simplified. Moreover, the quartic equation solution algorithm according to the second embodiment requires solution of the quadratic equation only three times, so that the volume of calculations is reduced as compared to the related-art solution algorithm.

Third embodiment

In the error correction steps constituting the error correction method according to the first and second embodiments, the error location is calculated by referring to the table in the Galois field. Alternatively, the error location is calculated using the construction described below.

Fig. 11 is a flowchart showing an error location calculation algorithm according to the third embodiment. Referring to Fig. 11, ST101 indicates a step for setting an initial value of an element S of a Galois field and an integer k. ST102 indicates a step for determining whether the algorithm may be terminated. ST103 indicates a step for subjecting the element S of the Galois field to comparison. ST104 indicates a step for updating the element S and the integer k. ST105 indicates a step for calculating an error location LOC.

A description will be given of the operation.

A detailed description will be given using a (n, k) three-bit correcting BCH code over a Galois field GF(2^8) having a code length of n for k symbols.

In step ST101, the root KON of the error location polynomial is substituted into the element S of the Galois field and 0 is substitute into the integer k.

In step ST102, a determination is made as to whether k is smaller than the code length n. If an affirmative answer is yielded in step ST102, the algorithm proceeds to step ST103. If a negative answer is yielded in step ST102, the process is terminated.

In step ST103, a determination is made as to whether S is equal to one of Galois field elements $\alpha^1(0 \leq l < 8)$. If an affirmative answer is yielded in step ST103, a sum of k and 1 is stored in the error location LOC in step ST105 and the process is terminated.

If a negative answer is yielded in step

ST103, the algorithm proceeds to step ST104.

In step ST104, the Galois field element S is multiplied by α^{-8}, 8 is added to the integer k, and the steps subsequent to step ST102 is repeated.

According to the third embodiment, the Galois field table for calculating an error location from the root of the error location polynomial is unnecessary. When the third embodiment is applied to the first and second embodiments, further reduction in the required table size is achieved. In the above description of the third embodiment, it is assumed that the error location is searched for every eight bits. In general, the error location may be searched for every n bits.

Fourth embodiment

Fig. 12 is a block diagram showing an error correction apparatus using a three-bit correcting BCH code according to a fourth embodiment the present invention. Referring to Fig. 12, the error correction apparatus comprises a syndrome generating circuit for calculating syndromes from a received word, and an error bit count estimating circuit for estimating the number of bits in error from the calculated syndromes. The syndrome generating circuit 1 and the error bit count estimating circuit 2 constitute error bit count estimating means.

The apparatus further comprises an error location polynomial generating circuit (polynomial generating means) for generating an error location polynomial in accordance with the number of bits in error estimated by the error bit count estimating circuit 2, an error location polynomial solution circuit 4 (polynomial solution means) for calculating the root of the error location polynomial generated by the error location polynomial generating circuit 3, an error location table 5, a delay circuit 6, a correcting circuit 7 for identifying an error location from the root of the error location polynomial calculated by the error location polynomial solution circuit 4, and correcting the bit value at the error location. The error location table 5, the delay circuit 6 and the correcting circuit 7 constitute correction means.

A description will now be given of the operation according to the fourth embodiment.

A detailed description will be given using a (n, k) three-bit correcting BCH code over a Galois field GF(2^8) having a code length of n for k symbols. It is assumed that the Galois field is processed on a polynomial base.

The received word is input to the syndrome generating circuit 1 and the delay circuit 6. The syndrome generating circuit 1 calculates syndromes S1, S3 and S5 from the received word and outputs the result of calculation to the error bit count estimating circuit 2.

The error bit count estimating circuit 2 estimates the number of bits in error occurring in the received word, based on the syndromes S1, S3 and S5 calculated by the syndrome generating circuit 1.

It is determined that there is no error if all of the syndromes S1, S3, and S5 are 0. The decoding process is then terminated. When not all of the syndromes S1, S3 and S5 are 0, a calculation $T=S1^3+S3$ is performed. If T=0, an estimation is made that there is a one-bit error. If T is not 0, an estimation is made that an error of two bits or more occurs.

When the error bit count estimating circuit 2 detects an error, the error location polynomial generating circuit 3 generates an error location polynomial. In the case of an error of two bits or more, the coefficients $\sigma1$, $\sigma2$, and $\sigma3$ of the error location polynomial (22) are generated. In the case of one-bit error, it is determined that $\sigma1=S1$, $\sigma2=0$, and $\sigma3=0$.

The error location polynomial solution circuit 4 calculates the root of the error location polynomial generated by the error location polynomial generating circuit 3. Before describing the detailed construction and operation of the error location polynomial solution circuit 4, a description will be given of a quadratic equation solution circuit 42, a cubic root calculation circuit 43, and a normalized cubic equation solution circuit 62 used in the error location polynomial solution circuit 4 (see Figs. 18 and 19).

Fig. 13 is a block diagram showing the quadratic equation solution circuit 42 (circuit for solving the quadratic equation $x^2+p\cdot x+q=0$). Referring to Fig. 13, the quadratic equation solution circuit 42 comprises a Galois field square circuit 11, a Galois field division circuit 12, a combinatorial circuit 13, a Galois field multiplication circuit 14 and a Galois field addition circuit 15.

First of all, coefficients p and q of the quadratic equation are input via the input terminal.

The Galois field square circuit 11 generates p^2 by raising p to the second power, and outputs p^2 to the Galois field division circuit 12.

The Galois field division circuit 12 generates $c=q/p^2$ by dividing q by p^2, and outputs $c=q/p^2$ to the combinatorial circuit 13.

The combinatorial circuit 13 is a linear combinatorial circuit given by expression (26). The combinatorial circuit 13 generates one root y of the normalized quadratic equation of expression (24) upon receiving $c=q/p^2$.

The Galois field multiplication circuit 14 multiplies p by y, and outputs one root $x1=p\cdot y$ of the quadratic equation to the output terminal and to the Galois field addition circuit 15.

The Galois field addition circuit 15 adds x1 and p and outputs the other root $x2=x1+p$ to the output terminal.

A description will now be given of the cubic root calculation circuit 43.

Fig. 14 is a block diagram showing the cubic root calculation circuit 43. Referring to Fig. 14, the cubic root calculation circuit 43 comprises a base translating circuit 21, subfield division circuits 22 and 27, translating circuits 23 and 26, lookup tables 24 and 28, a subfield addition circuit 25, a subfield multiplication circuit 29 and a base reverse translating circuit 30.

Fig. 15 is a block diagram showing the translating circuits 23 and 26. Referring to Fig. 15, each of the translating circuits 23 and 26 comprises a subfield square circuit 31, a subfield coefficient multiplication circuit 32, and subfield addition circuits 33 and 34. c indicates an element of the subfield.

A description will now be given of the operation according to the fourth embodiment.

It is assumed that a base over the subfield $GF(2^4)$ of the Galois field $GF(2^8)$ is $\{1, \beta\}$, where β indicates an element which does not belong to the subfield and satisfies $\beta^2+p\cdot\beta+q=0$ (p and q are elements of the subfield).

First, the translating circuits 23 and 26 of Fig. 15 will be described.

The subfield square circuit 31 for the subfield $GF(2^4)$ generates x^2 by raising x input thereto to the second power.

The subfield coefficient multiplication circuit 32 generates $p \cdot x$ by multiplying x by p.

The subfield addition circuit 33 generates $x^2+p \cdot x$ by adding x^2 and $p \cdot x$. The subfield addition circuit 34 generates $x^2+p \cdot x+c$ by adding $x^2+p \cdot x$ and constant c.

A description will now be given of the cubic root calculation circuit 43 shown in Fig. 14.

The base translating circuit 21 split an element B of the Galois field $GF(2^8)$ into elements of the subfield $GF(2^4)$, translating it into $B=b1 \cdot \beta+b0$, where b1 and b0 indicates elements of the subfield.

The subfield division circuit 22 calculates b0/b1, and outputs the calculation result to the translating circuit 23 and the subfield addition circuit 25.

The translating circuit 23 calculates b2 of expression (31) (constant c corresponds to q), and outputs the calculation result to the lookup table 24.

The lookup table 24 outputs one root y of expression (31) corresponding to b2 to the subfield addition circuit 25.

The subfield addition circuit 25 adds b0/b1 and y, generates the root x of expression (30), and outputs the root x to the translating circuit 26.

The translating circuit 26 generates $x^2+p \cdot x+p^2+q$ (constant c corresponds to p^2+q), and outputs $x^2+p\cdot x+p^2+q$ to the subfield division circuit 27.

The subfield division circuit 27 calculates $b1/(x^2+p\cdot x+p^2+q)$ and outputs the calculation result to the lookup table 28.

The lookup table 28 outputs the cubic root q of the subfield element $b1/(x^2+p\cdot x+p^2+q)$ to the subfield multiplication circuit 29 and the base reverse-translate circuit 30.

The subfield multiplication circuit 29 generates $x\cdot q$ by multiplying the root x output from the subfield addition circuit 25 by the cubic root q, and outputs $x\cdot q$ to the base reverse-translating circuit 30.

The base reverse translating circuit 30 generates $q\cdot\beta+x\cdot q$ by splitting the cubic root of B into the element of the subfield, fits $q\cdot\beta+x\cdot q$ to the original base so as to output the cubic root $B^{1/3}$ of B to the output terminal.

A description will now be given of the normalized cubic equation solution circuit 62 (circuit for solving the normalized cubic equation $x^3+r\cdot x+s=0$).

Fig. 16 is a block diagram showing the normalized cubic equation solution circuit 62. The normalized cubic equation solution circuit 63 comprises a Galois field cubic circuit 41, a quadratic equation solution circuit 42, a cubic root calculation circuit 43, a Galois field coefficient multiplication circuit 44, translating circuits 45 and 46, and a Galois field addition circuit 47.

Fig. 17 is a block diagram showing the translating circuits 45 and 46. Referring to Fig. 17, each of the translating circuits comprises a Galois field inverse element circuit 51, a Galois field multiplication circuit 52 and a Galois field addition circuit 53.

First, a description will be given of the translating circuits 45 and 46 of Fig. 17.

A Galois field element x input via the input terminal is input to the Galois field inverse element circuit 51 and the Galois field addition circuit 53. The first-order coefficient r of the normalized cubic equation is input to the Galois field multiplication circuit 52.

The Galois field inverse element circuit 51 generates an inverse element for x, i.e., 1/x and outputs 1/x to the Galois field multiplication circuit 52.

The Galois field multiplication circuit 52 generates r/x from 1/x and r and outputs the generated result r/x to the Galois field addition circuit 53.

The Galois field addition circuit 53 adds x and r/x and outputs x+r/x to the output terminal.

A description will now be given of the operation of the normalized cubic equation solution circuit 62 of Fig. 16.

First, the coefficients r and s of the normalized cubic equation are input via the input terminal.

The Galois field cubic circuit 41 receives r, raises r to the third power and outputs the result r^3 to the quadratic equation solution circuit 42.

When s and r^3 are input, the quadratic equation solution circuit 42 calculates one root x of the quadratic equation $x^2+s\cdot x+r^3=0$.

The cubic root calculation circuit 43 calculates the cubic root β of the root x calculated by the quadratic equation solution circuit 42, and outputs the cubic root β to the Galois field multiplication circuit 44 and the translating circuit 45.

The Galois field coefficient multiplication circuit 44 multiplies the cubic root β by the cubic root ω of 1, and outputs $\beta\cdot\omega$ to the translating circuit 46.

The translating circuits 45 and 46 calculate the two roots of expression (33) and output the roots to the output terminal. The calculated two roots are input to the Galois field addition circuit 47 and the third root is output to the output terminal.

Fig. 18 is a block diagram showing an error location polynomial solution circuit 4 (circuit for solving the error location polynomial $X^3+\sigma 1\cdot x^2+\sigma 2\cdot x+\sigma 3=0$). Referring to Fig. 18, the error location polynomial solution circuit 4 comprises a translating circuit 61, a normalized cubic equation solution circuit 62, and Galois field addition circuits 63-65.

A description will be given of the operation of the error location polynomial solution circuit 4.

The coefficients σ1, σ2, and σ3 of the third error location polynomial generated by the error location polynomial generating circuit 3 are input to the translating circuit 61. The coefficient σ1 is also input to the Galois field addition circuits 63-65.

The translating circuit 61 calculates the coefficients p and q of the normalized cubic equation of expression (6), and outputs the coefficients p and q to the normalized cubic equation solution circuit 62.

The normalized cubic equation solution circuit 62 calculates the three roots y1, y2 and y3 of $y^3+p\cdot y+q=0$ and outputs the three roots y1, y2 and y3 to the Galois field addition circuits 63-65.

The Galois field addition circuit 63 adds σ1 to y1 and outputs x1, the Galois field addition circuit 64 adds σ1 to y2 and outputs x2, and the Galois field addition circuit 65 adds σ1 to y3 and outputs x3.

The roots x1, x2, and x3 calculated by the error location polynomial solution circuit 4 are input to the error location table 5.

The error location table 5 stores the exponent of the element of the Galois field. Since the exponent of the element of the Galois field corresponds to the error location, the error location is identified by referring to the table with the calculated root (element of the Galois field) of the error location polynomial.

The correcting circuit 7 corrects the error in the received word stored in the delay circuit 6, and outputs the decoded result.

The error correction apparatus using a three-bit correcting BCH code according to the fourth embodiment is constructed such that the quadratic equation solution circuit 42 constituting the normalized cubic equation solution circuit 62 of the error location polynomial solution circuit 4 comprises Galois field operation circuits and simple combinatorial circuits. The cubic root calculation circuit 43 comprises subfield operation circuits and small-scale tables over the subfield. Thus, the circuit scale is reduced. Another advantage is that the control of the apparatus is simplified since the process does not allow branching according to the number of bits, by using the same sequence for respective number of bits.

Fifth embodiment

It is possible to construct the error correction apparatus using a three-bit correcting BCH code in a similar manner as the error correction apparatus using a four-bit correcting BCH code described in the fourth embodiment. The overall construction of the error correction apparatus using a four-bit correcting BCH code is similar to that of Fig. 12. A description of the operation of the error correction apparatus according to the fifth embodiment will be given with reference to Fig. 12.

The syndrome generating circuit 1 calculates syndromes S1, S3, S5 and S7 from the received word and outputs the syndromes S1, S3, S5 and S7 to the error bit count estimating circuit 2.

The error bit count estimating circuit 2 estimates the number of bits in error occurring in the received word, based on the syndromes S1, S3, S5 and S7 calculated by the syndrome generating circuit 1. It is determined that there is no error if all of the syndromes S1, S3, S5 and S7 are 0. The decoding process is then terminated. When not all of the syndromes S1, S3, S5 and S7 are 0, a calculation $V=S1(S1^5+S5)+S3(S1^3+S3)$ is performed. If V=0, an estimation is made that there is an error of two bits or fewer. If V is not 0, it is determined that there is an error of three bits or more.

When an error is detected by the error bit count estimating circuit 2, the error location polynomial generating circuit 3 generates an error location polynomial. That is, when an error of two bits or fewer is detected, the error location polynomial generating circuit 3 generates coefficients the σ1 and σ2 of the error location polynomial (42). When an error of three bits or more is detected, the coefficients σ1, σ2, σ3 and σ4 of the error location polynomial (43) are generated. When an error of two bits or fewer is detected, it is assumed that σ3=0 and σ4=0 so that the error location polynomial is processed as a quartic polynomial.

The quartic error location polynomial generated by the error location polynomial generating circuit 3 is input to the error location polynomial solution circuit 4.

Fig. 19 is a block diagram showing the error location polynomial solution circuit 4. Referring to Fig. 19, the error location polynomial solution circuit 4 comprises a quartic equation solution circuit 71, a quadratic equation solution circuit 72 and a selecting circuit 73.

Fig. 20 is a block diagram showing the quartic equation solution circuit 71. Referring to Fig. 20, the quartic equation solution circuit 71 comprises a coefficient translation circuit 81, Galois field division circuits 82, 83 and 85, a normalized cubic equation solution circuit 84, quadratic equation solution circuits 86, 89 and 90, Galois field multiplication circuits 87 and 88, and Galois field addition circuits 91-94.

Before describing the operation of the error location polynomial solution circuit 4 of Fig.

19, the operation of the quartic equation solution circuit 71 of Fig. 20 will be described.

The coefficients $\sigma1$, $\sigma2$, $\sigma3$, and $\sigma4$ input via the input terminal are translated by the coefficient translating circuit 81 into the coefficients p, q, r, and c defined in expression (44). c is input to the Galois field addition circuits 91-94. p is input to the Galois field division circuits 82 and 85, q is input to the Galois field division circuit 83, and r is input to the quadratic equation solution circuit 86 and the Galois field division circuits 82 and 83.

The Galois field division circuit 82 calculates p/r and outputs it to the normalized cubic equation solution circuit 84. The Galois field division circuit 83 calculates q/r and outputs it to the normalized cubic equation solution circuit 84.

The normalized cubic equation solution circuit 84 calculates one root λ of the normalized cubic equation (49). λ is output to the Galois field division circuit 85, and the Galois field multiplication circuits 87 and 88.

The Galois field division circuit 85 calculates p/λ and outputs it to the quadratic equation solution circuit 86.

The quadratic equation solution circuit 86 calculates the two roots q1 and q2 of the quadratic equation (50). q1 is input to the Galois field multiplication circuit 87 and the quadratic equation solution circuit 89. q2 is input to the Galois field multiplication circuit 88 and the quadratic equation solution circuit 90.

The Galois field multiplication circuit 87 calculates p1=λ·q1 and outputs it to the quadratic equation solution circuit 89.

The Galois field multiplication circuit 88 calculates p2=λ·q2 and outputs it to the quadratic equation solution circuit 90.

The quadratic equation solution circuit 89 calculates the two roots y1 and y2 of the quadratic equation (51). y1 and y2 are output to the Galois field addition circuits 91 and 92.

The quadratic equation solution circuit 90 calculates the two roots y3 and y4 of the quadratic equation (52). y3 and y4 are output to Galois field addition circuits 93 and 94.

The Galois field addition circuits 91-94 add c to y1, y2, y3, and y4, respectively and outputs the roots x1, x2, x3, and x4 of the quartic polynomial (43).

A description will be given of the operation of the error location polynomial solution circuit 4 of Fig. 19.

The coefficients σ1, σ2, σ3, and σ4 input via the input terminal are input to the quartic equation solution circuit 71. The coefficients σ1 and σ2 are input to the quadratic equation solution circuit 72.

The quartic equation solution circuit 71 calculates the four roots y1, y2, y3, and y4 of the quartic error location polynomial (43) and outputs y1, y2, y3, and y4 to the selecting circuit 73.

The quadratic equation solution circuit 72 calculates the two roots z1 and z2 of the quadratic error location polynomial (42) and outputs z1 and z2 to the selecting circuit 73.

In the case of an error of three bits or more, the selecting circuit 73 outputs x1=y1, x2=y2, x3=y3, and x4=y4. In the case of an error of two bits or fewer, It outputs x1=z1, x2=z2, x3=0, and x4=0.

The root calculated by the error location polynomial solution circuit 4 is input to the error location table 5. In a similar configuration as the fourth embodiment, the table is referred to with the calculated root (element of the Galois field) of the error location polynomial so as to identify the error location.

The correcting circuit 7 corrects the error in the received word stored in the delay circuit 6 and outputs the result of decoding.

With the above-described configuration, the error correction apparatus using a four-bit correcting BCH code according to the fifth embodiment provides an advantage in that the table size and the storage capacity are reduced. Since only two types of polynomials, i.e. the quadratic polynomial and the quartic polynomial, are dealt with, the control of the error correction apparatus is simplified.

Sixth embodiment

In the error correction apparatus according to the fourth and fifth embodiments, the root of the error location polynomial is used as a key in referring to the error location table to identify the error location. Alternatively, the error location may be identified using the configuration as described below.

Fig. 21 is a block diagram showing the construction of the error correction apparatus according to the sixth embodiment. In Fig. 21, those elements that are identical to the corresponding elements of Fig. 12 are designated by the same reference numerals so that the description thereof is omitted. The error correction apparatus according to the sixth embodiment comprises an error location detecting circuit 8.

Fig. 22 is a block diagram showing the error location detecting circuit 8. Referring to Fig. 8, the error location detecting circuit 8 comprises a counter 101, a storage circuit 102 for storing elements of a Galois field, a Galois field coefficient multiplication circuit 103, a comparing circuit 104 and an integer addition circuit 105.

The operation other than that of the error location detecting circuit 8 is the same as the corresponding operation according to the fourth and fifth embodiments so that the description below only concerns the error location detecting circuit 8.

The root (element of the Galois field) of the error location polynomial calculated by the error location polynomial solution circuit 4 is input to the error location detecting circuit 8.

The input Galois field element is input to the storage circuit 102 for storage therein. An initial value 0 is stored in the counter 101 before the operation is started and an upward counting operation proceeds in units of 8 at each clock.

The Galois field element stored in the storage circuit 102 is input to the comparing circuit 104 and the Galois field coefficient multiplication circuit 103.

The comparing circuit 104 compares the input Galois field element with each of the Galois field elements $\{\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\}$. If the input Galois field element is determined to match α^k, the comparing circuit 104 outputs k to the integer addition circuit 105.

The Galois field coefficient multiplication circuit 103 multiplies the Galois field element stored in storage circuit 102 by α^{-8} and outputs the result to the storage circuit 102.

The integer addition circuit 105 adds the input value from the comparing circuit 104 and the input value from the counter 101 and outputs the sum as the error location.

If there are a plurality of non-zero roots of the error location polynomial, the above-described steps may proceed for each root.

With the above-described configuration, the error correction apparatus according to the sixth embodiment eliminates the need for an error location table for identifying the error location. Thus, the circuit scale is further reduced. By arranging more than two error location detecting circuits 8 in parallel, the processing speed is improved. For example, three error location detecting circuits 8 may be provided for a three-bit correcting BCH code and four error location detecting circuits 8 may be provided for a four-bit correcting BCH code. With this, waiting during the process is eliminated.

In the fourth through sixth embodiment, the error correction is implemented by hardware means. Alternatively, computer software (error correction program) capable of the operation of the error correction apparatus may be prepared and stored in a computer readable recording medium.

Seventh embodiment

In the first through sixth embodiments, the syndromes S1, S3 and S5 are calculated from the received word. Alternatively, the Galois subfield arithmetic operation may be performed so as to calculate the syndromes from the received word.

A specific description will be given of

error correction using a three-bit correcting BCH code. In decoding a BCH code, the Galois field operation is necessary. In the seventh embodiment, the arithmetic operation in the Galois field GF(2^{2m}) is reduced to the arithmetic operation in its subfield GF(2^m). Hereinafter, the Galois field GF(2^{2m}) is denoted by K and the subfield GF(2^m) is denoted by L. α denotes a primitive element of the Galois field K and γ denotes a generator of the subfield L. A relation $\gamma=\alpha^l$ (l is an integer) is satisfied. The base over the subfield L of the Galois field K is denoted by {1, β}. β indicates an element of K that does not belong to L. β satisfies a relation $\beta^2+p\beta+q=0$ (p and q are elements of L). In general, an element X of K is represented as $X=x1\beta+x0$ using elements x1, x2 of L. In this specification, $x1\beta+x0$ is represented as (x1, x0).

Operations in the subfield L may be represented using a polynomial base, a normalized base or an exponent. Given below is a description of the operation using the exponential expression.

Given that the subfield L is represented using the exponent expression, an element γ^i of the subfield L is represented by an exponent i (i=0, 1, ..., 2^m-2). In this case, a product $\gamma^i\cdot\gamma^j$ of the two elements of L is calculated in the form of $i+j(\mathrm{mod}2^m-1)$. Division γ^i/γ^j is calculated in the form of $i-j(\mathrm{mod}2^m-1)$. Addition is performed using Zech logarithm. Zech logarithm is logarithm Z[*] that satisfies $1+\gamma^j=\gamma^{Z[j]}$. Using Zech logarithm, an addition $\gamma^i+\gamma^j$ is calculated in the form of $i+Z[j-i](\bmod 2^m-1)$, because $\gamma^i+\gamma^j=\gamma i(1+\gamma^{j-i})=\gamma^{i+Z[j-i]}$.

A method of operation in the Galois field $K=GF(2^{2m})$ will be described.

An addition X+Y of two elements X=(x1,x0) and Y=(y1,y0) of K, where x0, x1, y0, and y1 indicates elements of the subfield L, is performed.

$$X+Y=(x1+y1,\ x0+y0) \quad (61)$$

The product X·Y of X and Y is calculated according to the expression below.

$$X\cdot Y=(x1\cdot y0+x0\cdot y1+p\cdot x1\cdot y1, x0\cdot y0+q\cdot x1\cdot y1) \quad (62)$$

An inverse element X^{-1} of X is calculated according to the expression below.

$$X^{-1}=(x1\cdot w^{-1}, (x0+x1)\cdot w^{-1}) \quad (63)$$
$$w=x0\cdot(x0+x1)+q\cdot x1^2$$

A division X/Y is performed as a combination of an inverse element and multiplication.

The calculation of Galois field K multiplication and inverse element will be described with reference to the drawings.

In the following description, it is assumed that p=1. Fig. 27 is a flowchart showing how a product $X\cdot Y$ of the two elements $X=(x1, x0)$, $Y=(y1,y0)$ of the Galois field K is calculated. The calculation of expression (62) is divided into addition, multiplication and multiplication by a constant. Variables $z0$, $z1$ and $z2$ for storing elements of the subfield L are used in the calculation.

In step ST221, the two elements $X=(x1,x0)$,$Y=(y1,y0)$ of the Galois field K are set.

In step ST222, $x1\cdot y1$ is substituted for $z0$, $x1+x0$ is substituted for $z1$, and $y1+y0$ is substituted for $z2$.

In step ST223, a product $(z0\cdot q)$ of $z0$ and a constant q is substituted for $z0$, and $z1\cdot z2$ is substituted for $z1$. In this state, the content of $z1$ is $x0\cdot y0+x0\cdot y1+x1\cdot y0+x1\cdot y1$.

In step ST224, $x0\cdot y0$ is substituted for $z2$.

In step ST225, $z2$ is added to $z0$ and $z1$, a left element in the parentheses of expression (62) is substituted for $z1$, and a right element is substituted for $z0$.

In the last step ST226, a product $X\cdot Y=(z1,z0)$ is output.

Fig. 28 is a flowchart showing a calculation of an inverse element X^{-1} of the element $X=(x1, x0)$ of the Galois field K. The calculation of expression (63) is divided into addition, multiplication and multiplication by a constant. Variables $z0$, $z1$ and $z2$ for storing elements of the subfield L are used in the calculation.

In step ST231, the element X=(x1, x0) of the Galois field K is set.

In step ST232, x1+x0 is substituted for z0 and z1, and x1·x1 is substituted for z2.

In step ST233, a product (z2·q) of z2 and a constant q is substituted for z2, and x0·z1 is substituted for z1.

In step ST234, z1+z2 is substituted for z2.

In step ST235, z0/z2 is substituted for z0, and x1/z2 is substituted for z1. In this state, a left element in the parentheses of expression (63) is stored in z1, and a right element is stored in z0.

In the last step ST236, an inverse element X^{-1}=(z1, z0) is output.

Fig. 23 is a flowchart of an error correction method using a three-bit correcting BCH code, comprising step ST201 for calculating syndromes from a received word by performing Galois subfield arithmetic operations, step ST202 for determining whether there is an error, step ST203 for terminating the process upon determination that there is no error, step ST204 for determining whether there is a one-bit error, step ST205 for calculating the root of an error location polynomial upon determination that there is a one-bit error, step ST206 for calculating a cubic error location polynomial, step ST207 for calculating three roots of the cubic error location polynomial, step ST208 for calculating the location of a bit in error, based on the roots of the error location polynomial, and step ST209 for correcting the bit in error.

A detailed description of the operation shown in Fig. 23 will be given using a three-bit correcting BCH code having a code length of n.

In step ST201, the syndromes S1, S3 and S5 are calculated from the received word.

Expressing the received bits (r_{n-1}, r_{n-2}, ...,r_1, r_0) by the polynomial shown below, the syndromes are calculated as $S1=R(\alpha)$, $S3=R(\alpha^3)$, and $S5=R(\alpha^5)$, where α indicates a primitive element of the Galois field K.

$$R(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+..+r_2x^2+r_1x+r_0 \quad (64)$$

The syndromes S1, S3, and S5 are calculated by using the operation in the Galois field described above. Fig. 24 is a flow chart showing a method of calculation of the syndrome S1. (a1,a0) in Fig. 24 is a subfield representation of the primitive element α of the Galois field K. Referring to Fig. 24, step ST211 is an initial value setting step. x0 and x1 indicate variables storing elements of the subfield L, and k indicates a variable storing an integer. ST212 indicates a step for processing received bits, ST213 indicates a step for determining a condition, ST214 indicates a step for updating the variables, ST215 indicates a step for storing the syndrome S1.

In step ST211, 0 is set in x0 and x1. 0 indicates a zero element of the subfield. n-1 (code length-1) is set in the variable k.

In step ST212, the variable x0 and the received bit r_k are added, and $x0+r_k$ is stored in the variable x0. The received bit 0 and 1 are processed as a zero element 0 and a unit element 1 of the subfield.

In step ST213, a check is made to determine whether k is 0, proceeds to step ST215 if an affirmative answer is yielded in step ST213, and proceeds to step ST214 if a negative answer is yielded in step ST213.

In step ST214, a product (a1, a0)·(x1, x0) of α =(a1, a0) and (x1, x0) is substituted for a combination of variables (x1, x0). A product (a1, a0)·(x1, x0) is calculated according to expression (62). k is decreased and the control returns to step ST212.

In step ST215, (x1, x0) is substituted for the syndrome S1 and the process is terminated.

The syndrome $S1=R(\alpha)$ is calculated according to the flow chart of Fig. 24. The syndromes S3 and S5 are also calculated in a manner similar to S1. A difference is that, in calculating the syndrome S3, α^3=(b1, b0) is used instead of α=(a1, a0), and, in calculating the syndrome S5, α^5=(c1, c0) is used in stead of α=(a1, a0). b0, b1, c0, c1 indicate elements of the subfield L.

The syndromes S1, S3, and S5 calculated in step ST201 are represented as S1=(x1, x0), S3=(y1, y0), and S5=(z1, z0).

In step ST202, if all of the syndromes S1, S3 and S5 are 0, i.e., if x0, x1, y0, y1, z0, z1 are all 0, it is determined that there is no error so that the decoding process is terminated (step ST203). Otherwise, $T=S1^3+S3=(x1,\ x0)^3+(y1,\ y0)$ is calculated in step ST204. If T=0, it is determined that there is a one-bit error so that the control is turned to step ST205.

In step ST205, S1=(x1, x0) is set as the root X of the error location polynomial X+S1, and the control is turned to step ST208.

If T is not 0, the coefficients σ1, σ2, and σ3 of the cubic error location polynomial shown below are calculated from the syndromes S1, S3, and S5 (step ST206). The coefficients σ1, σ2, and σ3 of the expression below are also calculated using the subfield L operation.

$$X^3+\sigma1\cdot x2+\sigma2\cdot x+\sigma3=0 \quad (65)$$

$$\sigma1=S1$$

$$\sigma2=S1^2+(S1^5+S5)/(S1^3+S3)$$

$$\sigma3=S3+S1\cdot(S1^5+S5)/(S1^3+S3)$$

Expression (65) is transformed into a normalized cubic equation (66) by defining such that X=Y+σ1.

$Y^3+A\cdot Y+B=0$, $A=\sigma 1^2+\sigma 2$, and $B=\sigma 1\cdot\sigma 2+\sigma 3$ (66)

Expression (66) is transformed into the following expression by defining such that $Y=Z+A/Z$.

$Z^6+B\cdot Z^3+A^3=0$ (67)

Expression (67) is a quadratic equation of Z^3. A description will now be given of a solution of a quadratic equation over the Galois field K by using the subfield L operation. A general quadratic equation over the Galois field K is given by expression (68). By defining such that $X=C\cdot Y$, expression (68) is transformed into a normalized expression (69).

$X^2+C\cdot X+D=0$ (68)

$Y^2+Y+E=0$ (69)

$E=D/C^2$

By defining such that $Y=(y1, y0)$ and $E=(e1, e0)$, expression (69) is transformed into expression (70) and expression (71).

$p\cdot y1^2+y1+e1=0$ (70)

$y0^2+y0+e0+q\cdot y1^2=0$ (71)

By defining such that $y1=z/p$, expression (70) is transformed into expression (72).

$z^2+z+p\cdot e1=0$ (72)

Expression (72) is a normalized quadratic equation over the subfield L, and, by referring to the table storing roots for respective constant terms, one root z is obtained. Thus, one root $y1=z/p$ of expression (70) is obtained, and the constant term $e0+q\cdot y1^2$ of expression (69) is determined. The above-mentioned table is referred to again with the constant term thus determined so as to obtain one root y0 of the normalized quadratic equation (71). The steps described above provide one root $Y=(y1,\ y0)$ of expression (69). The other root of expression (69) is $Y+1=(y1,\ y0+1)$, and the two roots of expression (68) are calculated by $X=C\cdot Y$ and $C\cdot Y+C$.

One root $Z^3=C$ of expression (67) is calculated by using the solution of quadratic equation mentioned above. If the cubic root of C is calculated by using the method described in the first embodiment, the root $Z=C^{1/3}$, $C^{1/3}\Omega$, and $C^{1/3}\Omega$ 2 of expression (67) is calculated, where Ω indicates a cubic root of the unit element of the Galois field K. The three roots Y1, Y2 and Y3 of expression (66) are calculated by $Y=Z+A/Z$.

$Y1=C^{1/3}+A/C^{1/3}$

$Y2=C^{1/3}\cdot\Omega+A/(C^{1/3}\cdot\Omega)$

$Y3=C^{1/3}\cdot\Omega^2+A/(C^{1/3}\cdot\Omega^2)=Y1+Y2$ (73)

In addition, the three roots $X1=Y1+\sigma 1$, $X2=Y2+\sigma 1$ and $X3=Y3+\sigma 1$ of expression (65) are calculated according to $X=Y+\sigma 1$.

In step 208, the location of the bit in error is calculated from the root $X=(x1, x0)$ (element of Galois field K) of the error location polynomial. When the root X of the error location polynomial is represented as $X=\alpha^{i}$ by using the primitive element α of K, i corresponds to the location of the bit in error. A description will be given below of a method of calculating the location i of the bit in error.

Since x0 and x1 are elements of the subfield L, x0 and x1 are represented such that $x0=\gamma^{j0}$ and $x1=\gamma^{j1}$, respectively, by using the generator γ of the subfield L, where j0 and j1 indicate suitable integers. X is transformed into expression (74).

$$X=\gamma^{j1}\beta+\gamma^{j0}=\gamma^{j1}(\beta+\gamma^{j0-j1}) \quad (74)$$

A table T[*], which satisfies the relation of expression (75), is prepared.

$$\alpha^{T(j)}=\beta+\gamma^{j} \quad (75)$$

If the table T[*] of expression (75) is used, expression (74) is transformed into the expression below.

$$X=\gamma^{j1}\alpha^{T[j0-j1]}=\alpha^{l\times j1+T[j0-j1]} \quad (76)$$

Here, the relatioOn $\gamma=\alpha^{l}$ is used. The location i=l*j1+T[j0-j1] of the bit in error is determined from expression (76).

When the roots of the error location polynomial include 0, there is no corresponding error location. In this case, the calculation of the error location is not performed (the cubic error location polynomial has the root 0 when there is a two-bit error).

In step ST209, the bit at the error location i calculated in step ST208 is inverted so that the error is corrected.

Thus, according to the seventh embodiment, the location of the bit in error is calculated and the error is corrected at a high speed, by directly solving the error location polynomial. Since all Galois field K operations are performed as operations in the subfield L, the operation table is eliminated. In the description above, it is assumed that operations are performed using the exponential expression in the subfield L. While the storage volume required to store the Zech logarithm of the extended field K is 2^{2m} words by 2m bits, the storage volume of the Zech logarithm table of the subfield is 2^{m} words by m bits. Thus, by using calculation using the subfield as described above, significant reduction in the storage volume is achieved. The representation in the subfield L is not necessarily an exponential expression. Vector representation, normal base and dual basis may also be used.

In the seventh embodiment, it is disclosed that a three-bit correcting BCH code is used. Alternatively, a one-bit correcting BCH code (Hamming code or extended Hamming code), a two-bit correcting BCH code, or a BCH code capable of correcting four bits or more may also be used.

Eighth embodiment

The error correction method as described in the seventh embodiment may also be implemented by software.

Fig. 25 is a block diagram showing an error correction apparatus using a one-bit correcting BCH code according to the eighth embodiment. The error correction apparatus according to the eighth embodiment comprises an input and output interface (hereinafter, referred to as I/O) for controlling the input and output of the received word, a memory (hereinafter, referred to as RAM) 202 for storing received words and variables for decoding, a memory (hereinafter, referred to as ROM) 203 for storing programs for the Galois field arithmetic operation and the decoding algorithm, a CPU 204 for reading the program from the ROM 203, executing the Galois field arithmetic operation algorithm and the decoding algorithm, and controlling the blocks, and an internal bus 205.

A description will now be given of the operation according to the eighth embodiment.

The received word is stored in the RAM 202 via the I/O 201.

The CPU 204 then reads the program for calculating the syndrome S1 from the ROM 203.

The program for calculating the syndrome S1 is executed as shown in the flowchart of Fig. 24. First, variables x0 and x1 for storing the syndrome S1 are placed in the RAM 202 so as to store an initial value 0. n-1 is set in a counter (not shown) in the CPU 204 (the above steps correspond to step ST211 of Fig. 24).

The received bit stored in the RAM 202 is then read and stored in a register (not shown) in the CPU 204. The content of this register and the variable x0 are added and a sum thereof is stored in the variable x0 (corresponding to step ST212 of Fig. 24).

A determination is then made as to whether the value of the counter is 0. If it is 0, the process is terminated. If the value of the counter is not 0, the count is decreased, the multiplication program, which uses the subfield of the Galois field is read from the ROM 203, and a product (a1, a0)·(x1, x0) of (a1, a0) and (x1, x0) is calculated. The multiplication program comprises the steps shown in the flowchart of Fig. 27. The multiplication result is stored in (x1, x0) and the process returns to a processing routine of step ST212 shown in Fig. 24. The above steps are repeated until the count reaches 0.

When the calculation of the syndrome S1=(x1, x0) is completed, a determination is made as to whether there is an error. If the syndrome S1 is 0, a determination is made that there is no error so that the decoding process is terminated. Otherwise, S1=(x1, x0) is set as the root X1 of the error location polynomial X+S1.

By storing the table defined by expression (75) in the ROM 203, the location of the error bit is calculated from the root X=(x1, x0) of the error location polynomial, using the method described in the seventh embodiment. The bit at the calculated error location is read from the RAM 202, and the inverted bit is returned to the RAM 202. When the above-mentioned decoding program has been executed, the received word is output from the RAM 202 via the I/O 201.

Thus, according to the eighth embodiment, Galois field operations are processed in an operation system of the subfield which is smaller than the original Galois field, so that the storage volume of the ROM 203 is reduced. By directly solving the error location polynomial to obtain the roots, the error location is calculated and the error is corrected at a high speed. The description above assumes the use of a one-bit correcting BCH code. Alternatively, the error correction apparatus of the eighth embodiment may easily be extended to a BCH code capable of correcting two bits or more.

Ninth embodiment

It is disclosed in the eighth embodiment that the program for the Galois field arithmetic operation algorithm and the decoding algorithm is stored in the ROM 203. Alternatively, the program may be supplied externally via the I/O 201. The program may be loaded from a floppy disk, a hard disk, an optical disk, a magneto-optical disk, a CD-ROM, a magnetic tape, or a non-volatile memory card.

Execution of the program may take place on an operating system or application software instead of a system CPU.

Any computer-readable recording medium may store the program. The program itself constitutes the invention.

The tenth embodiment

In the eighth embodiment, the error correction apparatus using a one-bit correcting BCH code is implemented by software. A portion of or the entirety of the algorithm may be implemented by hardware. It is especially to be noted that implementing the Galois field operation algorithm on hardware significantly reduced a delay in decoding.

Fig. 26 is a block diagram showing the error correction apparatus according to the tenth embodiment. Those components that are identical to the corresponding components of Fig. 25 are designated by the same reference numerals so that the description thereof is omitted. The error correction apparatus according to the tenth embodiment comprises a Galois field operation processor 206 comprising subfield operation circuits.

Fig. 29 is a block diagram of the Galois field operation processor 206. Referring to Fig. 29, the Galois field operation processor 206 comprises an instruction decoder 301 for decoding an instruction from a CPU, registers 302-305 for storing two input elements of the Galois field K (since an element of K is represented by two elements of the subfield L, four registers for storing subfield elements are provided), a constant q 306 of the subfield, registers 307-309 for temporarily storing interim results (subfield elements) of the calculation, and a subfield operation circuit system 310 using exponential representations of the subfield.

There are further provided exponential addition circuits 310A and 310C, an exponential subtraction circuit 310B, and a Zech logarithm table of the subfield 310D. Reference numeral 311 indicates an input selector for receiving x0, x1, y0, y1, z0, z1, z2, and the constant q from the registers 302-305 and supplying the same to the subfield operation circuit system 310, 312 represents a switch for controlling the input to the registers 307, 308 and 309 in accordance with the output from an output selector 313, and 313 indicates an output selector for selecting the output from the subfield operation circuit system 310.

The subfield operation circuit system 310 receives outputs a and b of the input selector 311 and performs multiplication, division, addition in the subfield using exponential expressions.

The exponential addition circuit 310A performs addition of a and b, and outputs a sum a+b to the output selector 313.

The exponential subtraction circuit 310B subtracts b from a, and outputs the result a-b to the subfield Zech logarithm table 310D and the output selector 313.

The subfield Zech logarithm table 310D outputs the Zech logarithm Z[a-b] corresponding to the input a-b to the exponential addition circuit 310C.

The exponential addition circuit 310C adds the input b and the output Z[a-b] from the Zech logarithm table 310D and supplies the sum b+Z[a-b] to the output selector 313.

The input c, d, e of the output selector 313 are as given by expression (77). c corresponds to subfield multiplication, d corresponds to subfield division and e corresponds to subfield addition.

$$c=a+b(\mathrm{mod}2^{m}-1)$$
$$d=a-b(\mathrm{mod}2^{m}-1)$$
$$e=b+Z[a-b](\mathrm{mod}2^{m}-1) \quad (77)$$

The Galois field operational processor 206 performs addition, multiplication, and division in the Galois field K in accordance with the CPU instruction.

First, a description will be given of an addition X+Y of two elements X=(x1, x0), Y=(y1,y0) of the Galois field K. x0 of X=(x1,x0) is stored in the register 302, and x1 is stored in the register 303. y0 of Y=(y1,y0) is stored in the register 304, and y1 is stored in the register 305.

At the first stage, the input selector 311 selects x0 as the output a and selects y0 as the output b. The output selector 313 selects e, and the switch 312A is closed so that the output e selected by the output selector 313 is provided to the register 307. The other switches 312B and 312C remain open.

In the second stage, the input selector 311 selects x1 as the output a and selects y1 as the output b. The output selector 313 selects e, and the switch 312B is closed so that the output e selected by the output selector 313 is provided to the register 308. The other switches 312A and 312C remain open.

Subsequently, a pair of register contents (z1, z0) are output as the result of operation. A description will now be given of a product X·Y of two elements X=(x1, x0),Y=(y1, y0). First, x0 of X=(x1, x0) is stored in the register 302 and x1 is stored in the register 303. y0 of Y=(y1, y0) is stored in the register 304, and y1 is stored in the register 305.

At the first stage, the input selector 311 selects x1 as the output a and selects y1 as the output b. The output selector 313 selects c, and the switch 312A is closed so that the output c selected by the output selector 313 is provided to the register 307. The other switches 312B and 312C remain open.

In the second stage, the input selector 311 selects x0 as the output a and selects x1 as the output b. The output selector 313 selects e, and the switch 312B is closed so that the output e selected by the output selector 313 is provided to the register 308. The other switches 312A and 312C remain open.

In the third stage, the input selector 311 selects y0 as the output a and selects y1 as the output b. The output selector 313 selects e, and the switch 312C is closed so that the output e selected by the output selector 313 is provided to the register 309. The other switches 312A and 312B remain open.

In the fourth stage, the input selector 311 selects z0 as the output a and selects q as the output b. The output selector 313 selects c, and the switch 312A is closed so that the output c selected by the output selector 313 is provided to the register 307. The other switches 312B and 312C remain open.

In the fifth stage, the input selector 311 selects z1 as the output a and selects z2 as the output b. The output selector 313 selects c, and the switch 312B is closed so that the output c selected by the output selector 313 is provided to the register 308. The other switches 312A and 312C remain open.

In the sixth stage, the input selector 311 selects x0 as the output a and selects y0 as the output b. The output selector 313 selects c, and the switch 312C is closed so that the output c selected by the output selector 313 is provided to the register 309. The other switches 312A and 312B remain open.

In the seventh stage, the input selector 311 selects z0 as the output a and selects z2 as the output b. The output selector 313 selects e, and the switch 312A is closed so that the output e selected by the output selector 313 is provided to the register 307. The other switches 312B and 312C remain open.

In the eighth stage, the input selector 311 selects z1 as the output a and selects z2 as the output b. The output selector 313 selects e, and the switch 312B is closed so that the output e selected by the output selector 313 is provided to the register 308. The other switches 312A and 312C remain open.

Subsequently, a pair of register contents (z1, z0) are output as the result of operation. A description will now be given of a calculation of an inverse element X^{-1} of the element X=(x1,x0) of the Galois field. x0 of X=(x1,x0) is stored in the register 302, and x1 is stored in the register 303.

At the first stage, the input selector 311 selects x0 as the output a and selects x1 as the output b. The output selector 313 selects e, and the switches 312A and 312B are closed so that the output e selected by the output selector 313 is provided to the registers 307 and 308. The switch 312C remains open.

In the second stage, the input selector 311 selects x1 as the output a and selects x1 as the output b. The output selector 313 selects c, and the switch 312C is closed so that the output c selected by the output selector 313 is provided to the register 309. The other switches 312A and 312B remain open.

In the third stage, the input selector 311 selects x0 as the output a and selects z1 as the output b. The output selector 313 selects c, and the switch 312B is closed so that the output c selected by the output selector 313 is provided to the register 308. The other switches 312A and 312C remain open.

In the fourth stage, the input selector 311 selects q as the output a and selects z2 as the output b. The output selector 313 selects c, and the switch 312C is closed so that the output c selected by the output selector 313 is provided to the register 309. The other switches 312A and 313B remain open.

In the fifth stage, the input selector 311 selects z1 as the output a and selects z2 as the output b. The output selector 313 selects e, and the switch 312C is closed so that the output e selected by the output selector 313 is provided to the register 309. The other switches 312A and 313B remain open.

In the sixth stage, the input selector 311 selects z0 as the output a and selects z2 as the output b. The output selector 313 selects d, and the switch 312A is closed so that the output d selected by the output selector 313 is provided to the register 307. The other switches 312B and 312C remain open.

In the seventh stage, the input selector 311 selects x1 as the output a and selects z2 as the output b. The output selector 313 selects d, and the switch 312B is closed so that the output d selected by the output selector 313 is provided to the register 308. The other switches 312A and 312C remain open. Subsequently, a pair of register contents (z1, z0) are output as the result of operation.

Thus, by using the Galois field operation processor 206, the arithmetic operation in the extended field is processed at a high speed. A large volume of multiplication and division operations are required in calculating the coefficients and roots of the error location polynomial. By using the processor of the tenth embodiment, a delay in decoding is significantly reduced. The syndromes are calculated at a high speed using the processor of the tenth embodiment.

By including the Galois field operation processor, the error correction apparatus according to the tenth embodiment is capable of processing the decoding at a high speed. The Galois field operation processor 206 flexibly processes the arithmetic operation in the extended field in accordance with an instruction from the CPU. Since the process comprises the subfield operation circuit system 310, the circuit scale is reduced as compared to a construction where the operation circuit for the extended field is used. In the tenth embodiment the exponential representation is used as a representation of the subfield. Alternatively, the vector representation, normal base and dual basis may also be used.

Eleventh embodiment

Calculation of the syndromes may be performed using the Galois field operation processor 206 described in the tenth embodiment. Alternatively, the operation may be implemented as circuit logic for further improvement in the decoding speed. Fig. 30 is a block diagram showing the error correction apparatus according to the eleventh embodiment. Referring to Fig. 30, those components that are identical to the corresponding components of Fig. 26 are designated by the same reference numerals so that the description thereof is omitted. Reference numeral 207 indicates a syndrome generating circuit.

Before describing the details of the syndrome generating circuit 207, a description will be given of the principle of the circuit according to the eleventh embodiment. As shown in the flowchart of Fig. 24, calculation of the syndrome S1 is based on a sum of products (a1, a0)· (x1,x0)+(0,r_k) composed of the constant α=(a1, a0) and the variables (x1, x0). Expression (62) shows that the product is given by (a1x0+a0x1+a1x1,a0x0+qa1x1). To simplify the representation, it is defined such that p=1. By going through a cycle of the flowchart of Fig. 24 one time, (a1+a0)·(x1+x0)+a0x0 is substituted for x1, and a0x0+qa1x1+r_k is substituted for x0. By defining such that a0=c0, q·a1=c1 and a1+a0=c2, the substitution is given by:

$$x0 \leftarrow c0 \cdot x0 + c1 \cdot x1 + r_k$$
$$x1 \leftarrow c2 \cdot (x1 + x0) + c0 \cdot x0 \quad (78)$$

These calculations are implemented by a subfield addition circuit, a c0 multiplication circuit for a subfield element, a c1 multiplication circuit for a subfield element and a c2 multiplication circuit for a subfield element. Fig. 31 is a block diagram of the syndrome generating circuit 207. Referring to Fig. 31, the syndrome generating circuit 207 comprises registers 401, 402 for storing elements of the subfield, a c0 multiplication circuit 403, a c1 multiplication circuit 404, a c2 multiplication circuit 405, an input terminal 406 for receiving bits, and subfield addition circuits 407-410.

A description will now be given of the operation of the syndrome generating circuit 207.

First of all, 0 is stored in the registers 401 and 402. 0 indicates a zero element of the subfield. The received bits r_k (k=0, 1, ..., n-1) are input bit by bit via the input terminal 406 starting with k=n-1 in the high to low order. In the following description, it is assumed that the reception proceeds until the (k+1)th bit is received.

The content (x0) of the register 401 is input to the c0 multiplication circuit 403 and the subfield addition circuit 410. The content (x1) of the register 402 is input to the c1 multiplication circuit 404 and the subfield addition circuit 410.

The c0 multiplication circuit 403 multiplies $x0$ by $c0$ and the result $c0 \cdot x0$ is input to the subfield addition circuit 407. The c1 multiplication circuit 404 multiplies $x1$ by $c1$ and the result $c1 \cdot x1$ is input to the subfield addition circuit 407. The subfield addition circuit 410 adds $x0$ to $x1$ and the result $x0+x1$ is input to the c2 multiplication circuit 405.

The subfield addition circuit 407 adds the input subfield element $c0 \cdot x0$ to $c1 \cdot x1$ and the result $c0 \cdot x0+c1 \cdot x1$ is input to the subfield addition circuit 408. The c2 multiplication circuit 405 multiplies the output $x0+x1$ from the subfield addition circuit 410 by $c2$ and the result $c2 \cdot (x0+x1)$ is input to the subfield addition circuit 409.

The subfield addition circuit 408 adds $c0 \cdot x0+c1 \cdot x1$ to the received bit r_k and the result $c0 \cdot x0+c1 \cdot x1+r_k c0 \cdot x0+c1 \cdot x1$ is input to the register 401. The subfield addition circuit 409 adds $c2 \cdot (x0+x1)$ to $c0 \cdot x0$ and the result $c2 \cdot (x0+x1)+c0 \cdot x0$ is placed in the register 402. This completes a cycle of the process. With this, the registers 401 and 402 indicated in expression (78) are updated. When the input of the entire received bits is complete, the content $x0$ of the register 401 and the content $x1$ of the register 402 are output as the syndrome $S1=(x1, x0)$.

Since the error correction apparatus

according to the eleventh embodiment is provided with a dedicated circuit for calculating the syndrome, the time required for decoding is significantly reduced.

INDUSTRIAL APPLICABILITY

The error correction method, the error correction apparatus and the error correction program according to the present invention are suitably used for correction of errors occurring in communication data and recorded data processed in digital radio communication and digital magnetic recording.